2608 Sweetgum Drive
Apex NC 27502
Toll-free: 800-549-9377
International: 919-387-0076
FAX: 919-387-1302

XS95 Board V1.3
User Manual

How to install, test, and use
your new XS95 Board

RELEASE DATE: 2/15/2000

Limited Warranty

X Engineering Software Systems Corp. (XESS) warrants that the Product, in the course of its normal use, will be free from defects in material and workmanship for a period of one (1) year and will conform to XESS's specification therefor. This limited warranty shall commence on the date appearing on your purchase receipt.

XESS shall have no liability for any Product returned if XESS determines that the asserted defect a) is not present, b) cannot reasonably be rectified because of damage occurring before XESS receives the Product, or c) is attributable to misuse, improper installation, alteration, accident or mishandling while in your possession. Subject to the limitations specified above, your sole and exclusive warranty shall be, during the period of warranty specified above and at XESS's option, the repair or replacement of the product. The foregoing warranty of XESS shall extend to repaired or replaced Products for the balance of the applicable period of the original warranty or thirty (30) days from the date of shipment of a repaired or replaced Product, whichever is longer.

THE FOREGOING LIMITED WARRANTY IS XESS'S SOLE WARRANTY AND IS APPLICABLE ONLY TO PRODUCTS SOLD AS NEW. THE REMEDIES PROVIDED HEREIN ARE IN LIEU OF a) ANY AND ALL OTHER REMEDIES AND WARRANTIES, WHETHER EXPRESSED OR IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, AND b) ANY AND ALL OBLIGATIONS AND LIABILITIES OF XESS FOR DAMAGES INCLUDING, BUT NOT LIMITED TO ACCIDENTAL, CONSEQUENTIAL, OR SPECIAL DAMAGES, OR ANY FINANCIAL LOSS, LOST PROFITS OR EXPENSES, OR LOST DATA ARISING OUT OF OR IN CONNECTION WITH THE PURCHASE, USE OR PERFORMANCE OF THE PRODUCT, EVEN IF XESS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

In the United States, some statutes do not allow exclusion or limitations of incidental or consequential damages, so the limitations above may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

1 Preliminaries

Getting Help!

Here are some places to get help if you encounter problems:

- If you can't get the XS95 Board hardware to work, send an e-mail message describing your problem to fpga-bugs@xess.com or check our web site at http://www.xess.com. Our web site also has
  - answers to frequently-asked-questions,
  - example designs for the XS Boards,
  - a place to sign-up for our email forum where you can post questions to other XS Board users.
- If you can't get your XILINX Foundation software tools installed properly, send an e-mail message describing your problem to hotline@xilinx.com or check their web site at http://support.xilinx.com.

Take notice!!

- The XS95 Board requires an external power supply to operate! It does not draw power through the downloading cable from the PC parallel port.
- If you are connecting a 9VDC power supply to your XS95 Board, please make sure the center terminal of the plug is positive and the outer sleeve is negative.
- The V1.3 version of the XS95 Board now uses a programmable oscillator with a default frequency of 50 MHz. You must reprogram the oscillator if you want to use another frequency. The procedure for doing this is described on page 7.

Packing List

Here is what you should have received in your package:

- an XS95 Board;
- a 6' cable with a 25-pin male connector on each end;
- a 3.5" floppy diskette or CDROM with software utilities and documentation for using the XS95 Board and documentation.

2
Installation

Installing the XSTOOLs Utilities and Documentation

XILINX currently provides the Foundation tools for programming their FPGAs and CPLDs. Any recent version of XILINX software should generate bitstream configuration files that are compatible with your XS95 Board. Follow the directions XILINX provides for installing their software.

XESS Corp. provides the additional XSTOOLs utilities for interfacing a PC to your XS95 Board. Run the SETUP.EXE program on the 3.5" diskette or CDROM to install these utilities.

Once the XSTOOLs are installed you will see the following subdirectories:

XSTOOLS\BIN contains the executable programs for downloading to the XS95 Board and for applying signals to the XS95 Board through the printer port. An assembler for the microcontroller on the XS95 Board is also included.

XSTOOLS\DOCS contains the documentation and schematics for the XS95 Board.

Applying Power to Your XS95 Board

You can use your XS95 Board in two ways, distinguished by the method you use to apply power to the board.

Using a 9VDC wall-mount

You can use your XS95 Board all by itself to experiment with logic and microcontroller designs. Just place the XS95 Board on a non-conducting surface as shown in Figure 1. Then apply power to jack J9 of the XS95 Board from a 9V DC wall transformer with a 2.1 mm female, center-positive plug. (See Figure 2 for the location of jack J9 on your XS95 Board.) The on-board voltage regulation circuitry will create the voltages required by the rest of the XS95 Board circuitry.

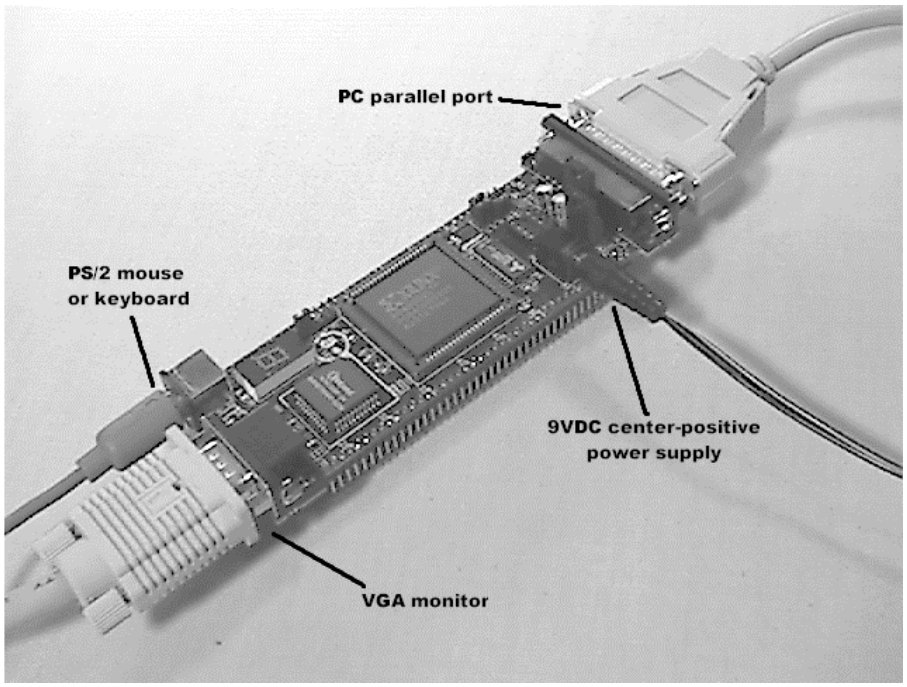
Solderless Breadboard Installation

The two rows of pins from your XS95 Board can be plugged into a solderless breadboard with holes spaced at 0.1" intervals. (One of the A.C.E. breadboards from 3M is a good choice.) Once plugged in, all the pins of the CPLD, microcontroller, and SRAM are accessible to other circuits on the breadboard. (The numbers printed next to the rows of pins on your XS95 Board correspond to the pin numbers of the CPLD.) Power can still be

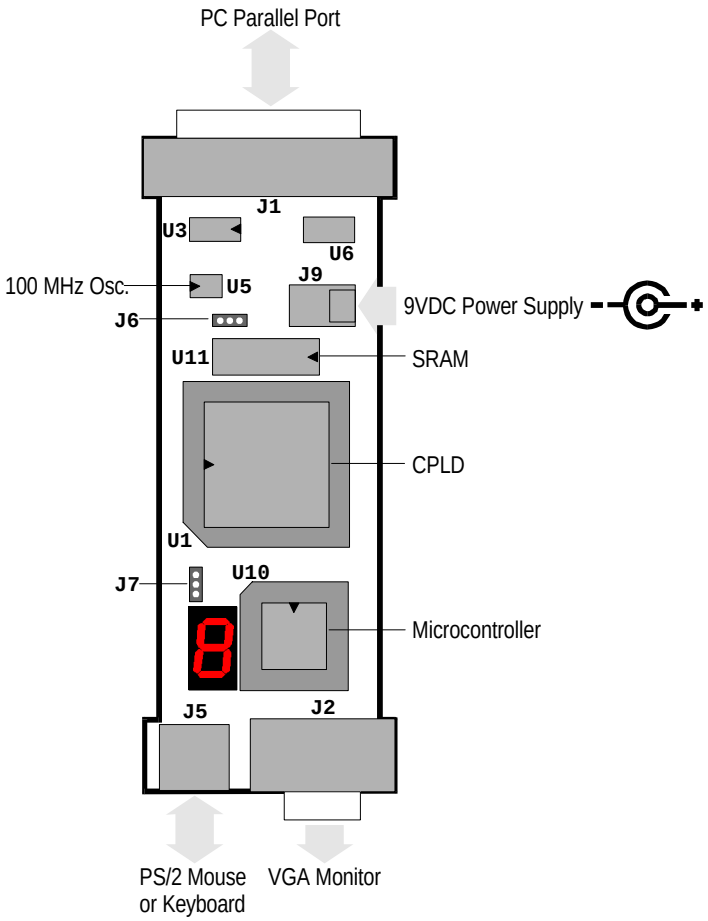
supplied to your XS95 Board though jack J9, or power can be applied directly through several pins on the underside of the board. Just connect +5V and ground to the following pins for your XS95 Board.

- Table 1: Power supply pins for the XS95 Board.

XS Board Type	GND Pin	+5V Pin
XS95-108 V1.3	49	78
XS95-108+ V1.3	49	78



- Figure 1: External connections to the XS95 Board.



• Figure 2: Arrangement of components on the XS95 Board.

Connecting a PC to Your XS95 Board

The 6' cable included with your XS95 Board connects it to a PC. One end of the cable attaches to the parallel port on the PC and the other connects to the female DB-25 connector (J1) at the top of the XS95 Board as shown in Figure 1.

Connecting a VGA Monitor to Your XS95 Board

You can display images on a VGA monitor by connecting it to the 15-pin J2 connector at the bottom of your XS95 Board (see Figure 1). You will have to download a VGA driver circuit to your XS95 Board to actually display an image. You can find an example VGA driver at http://www.xess.com.

Connecting a Mouse or Keyboard to Your XS95 Board

You can accept inputs from a keyboard or mouse by connecting it to the J5 PS/2 connector at the bottom of your XS95 Board (see Figure 1). You can find an example keyboard driver at http://www.xess.com.

Setting the Jumpers on Your XS95 Board

The default jumper settings shown in Table 2 configure your XS95 Board for use in a logic design environment. You will need to change the jumper settings only if you are:

- reprogramming the clock frequency on your XS95 Board (see page 7),
- executing microcontroller code from internal ROM instead of the external SRAM on the XS95 Board. (You will have to replace the ROMless microcontroller on the XS95 Board with a ROM version to use this feature.)

• Table 2: Jumper settings for XS95 Board.

Jumper	Setting	Purpose
J6	2-3 (osc) (default)	The shunt should be installed on pins 2 and 3 (osc) during normal operations when the programmable oscillator is generating a clock signal.
	1-2 (set)	The shunt should be installed on pins 1 and 2 (set) when the programmable oscillator frequency is being set.
J7	1-2 (ext) (default)	The shunt should be installed on pins 1 and 2 (ext) if the microcontroller program is stored in the external SRAM (U11) of the XS95 Board.
	2-3 (int)	The shunt should be installed on pins 2 and 3 (int) if the program is stored internally in the ROM of the microcontroller.

Testing Your XS95 Board

Once your XS95 Board is installed and the jumpers are in their default configuration, you can test it by executing the following command in a DOS window:

```
C:> XSTEST XS95-108
```

The test procedure programs the CPLD, loads the SRAM with a test program for the microcontroller, and then the microcontroller executes this program. The total test period (including programming the board) is about a minute for an XS95 Board. If the test completes successfully, then you will see a O displayed on the LED digit.

However, if the test program detects an error, then the LED digit displays an E or remains blank. In this case, check the following items:

- Make sure the XS95 Board is receiving power from a 9V DC power supply through jack J9 or through the VCC and GND pins.

- Check that the XS95 Board is sitting upon a non-conducting surface and that there are no connections to any of the pins (except for the VCC and GND pins if this is the way you are powering the board).

- Verify that the jumpers are in their default configuration.

- Make sure the downloading cable is securely attached to the XS95 Board and the PC parallel port.

- Verify that the parallel port is in ECP mode. (The mode is usually set in the BIOS as either SPP, EPP, ECP, or bidirectional. ECP mode works most reliably while bidirectional mode is not recommended.)

If all these checks are positive, then test the board using another PC. In our experience, 99.9% of all problems are due to the parallel port. If you cannot get your XS95 Board to pass the test even after taking these steps, then contact XESS Corp. to get a replacement board.

Programming the XS95 Board Clock Oscillator

The XS95 Board has a 100 MHz programmable oscillator (a Dallas Semiconductor DS1075Z-100). The 100 MHz master frequency can be divided by factors of 1, 2, ... up to 2052 to get clock frequencies of 100 MHz, 50 MHz, ... down to 48.7 KHz, respectively. The divided frequency is sent to the CPLD as a clock signal.

The divisor is stored in non-volatile memory in the oscillator chip so it will resume operation at its programmed frequency whenever power is applied to the XS95 Board. The following steps will store a particular divisor into the oscillator chip memory:

1) In a DOS window, use the following command with the particular clock divisor you want listed as the last argument:

```
C:\> XSSETCLK XS95-108 8
```

The example shown above will set the programmable oscillator on your XS95-108 Board to a frequency of 100 MHz / 8 = 12.5 MHz You may use any divisor between 1 and 2052 depending upon the clock frequency you want to use.

2) The XSSETCLK program will prompt you to remove the power and download cables from your XS95 Board. Then you should place a shunt at the **set** position of jumper J6. Then re-attach the download cable. Then reattach the power cable **only after the download cable is attached!**. When power is restored to the XS95 Board, the programmable oscillator will power up in its programming mode instead of generating a clock signal.

3) Press RETURN and the clock divisor will be programmed into the oscillator chip. If you wish to change the value of the divisor, you may re-issue the XSSETCLK command at this point with a new divisor without having to power-down the XS95 Board.

4) Finally, remove the power and download cables from your XS95 Board. Then place the shunt at the **osc** position of jumper J6. Then re-attach the download cable and the power cable. When power is restored to the XS95 Board, the programmable oscillator will power up in its active mode and output a clock signal at the programmed frequency.

3

Programming

This section will show you how to download a logic design from a PC into your XS95 Board.

Downloading Designs into Your XS95 Board

During the development and testing phases, you will usually connect your XS95 Board to the parallel port of a PC and download your circuit each time you make changes to it. You can download an XC9500-based design into the XS95 Board as follows:

```
C:\> XSLOAD CIRCUIT.SVF
```

where `CIRCUIT.SVF` is an XC9500 bitstream file that contains the configuration for the XC9500 CPLD. Make sure the file contains a bitstream for the type of XC9500 chip installed on your XS95 Board. This file is created using the XILINX Foundation software tools.

Use the following command if you need to configure the CPLD and also download an Intel-formatted HEX file into the static SRAM of the XS95 Board:

```
C:\> XSLOAD FILE.HEX CIRCUIT.SVF
```

where CIRCUIT.SVF is a bitstream file and FILE.HEX is a file containing hexadecimal data. The HEX file could contain microcontroller object code generated by the ASM51 assembler, or it could be arbitrary data from some other source. Whatever its source, the hexadecimal data is downloaded into the XS95 Board SRAM.

XSLOAD assumes the XS95 Board is connected to parallel port #1 of your PC. You can specify another port number using the -P option like so:

```
C:\> XSLOAD -P 2 FILE.HEX CIRCUIT.SVF
```

4

Programmer's Models

This section discusses the organization of components on the XS95 Board and introduces the concepts required to create applications that use both the microcontroller and the CPLD (field programmable logic device). Building CPLD-based designs is covered in detail in the *Practical Xilinx Designer Lab Book* by Prentice-Hall.

Microcontroller + CPLD Design Flow

The basic design flow for building microcontroller+CPLD applications is shown in Figure 3. Initially you have to get the specifications for the system you are trying to design. Then you have to determine what inputs are available to your system and what outputs it will generate.

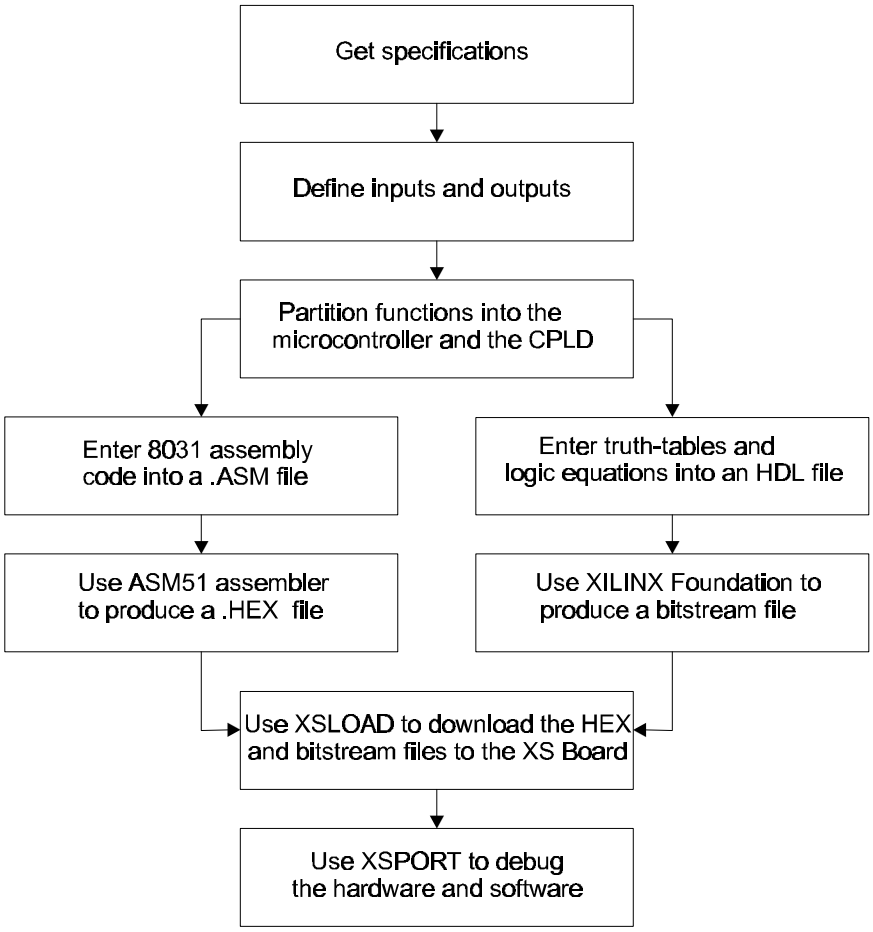
At this point, you have to partition the functions of your system between the microcontroller and the CPLD. Some of the input signals will go to the microcontroller, some will go to the CPLD, and some will go to both. Likewise, some of the outputs will be computed by the microcontroller and some by the CPLD. There will also be some new intra-system inputs and outputs created by the need for the microcontroller and the CPLD to cooperate.

In general, the CPLD will be used mainly for low-level functions where signal transitions occur more frequently and the control logic is simpler. A specialized serial transmitter/receiver would be a good example. Conversely, the microcontroller will be used for higher-level functions where the responses occur less quickly and the control logic is more complex. Reacting to commands passed in by the receiver is a good example. Once the design has been partitioned and you have assigned the various inputs, outputs, and functions to the microcontroller and the CPLD, then you can begin doing detailed design of the software and hardware. For the software, you can use your favorite editor to create a .ASM assembly-language file and assemble it with ASM51 to create a .HEX file for the microcontroller on the XS95 Board. For the CPLD hardware portion, you will enter truth-tables and logic equations into a .ABL or .VHDL file and compile it into an .SVF bitstream file using the XILINX Foundation software.

You can download the .HEX program file and the .SVF bitstream file to the XS95 Board using the XSLOAD program. XSLOAD stores the contents of the .HEX file into the SRAM on the XS95 Board and then it reconfigures the CPLD by loading it with the bitstream file.

When the XS95 Board is loaded with the hardware and software, you need to test it to see if it really works. The answer usually starts as "No" so you need a method of injecting test signals and observing the results. XSPORT is a simple program that lets you send test

signals to the XS95 Board through the PC parallel port. You can trace the reaction of your system to signals from the parallel port by programming the microcontroller and the CPLD to output status information on the LED digit (much like placing "printf" statements in your C language programs). This is admittedly crude but will serve if you don't have access to a programmable stimulus generator or logic analyzer.



• Figure 3: CPLD+microcontroller design flow.

XS95 Board Component Interconnections

The microcontroller and the CPLD on the XS95 Board are already connected together. These pre-existing connections save you the effort of having to wire them yourself, but they also impose limitations on how your microcontroller program and the CPLD hardware will interact. A high-level view of how the microcontroller, SRAM, and CPLD on the XS95 Board are connected is shown on the following pages. A more detailed schematic is also presented at the end of this manual.

The programmable oscillator output goes directly to a synchronous clock input of the CPLD. The CPLD uses this clock to generate a clock that it sends to the XTAL1 clock input of the microcontroller.

The microcontroller multiplexes the lower eight bits of a memory address with eight bits of data and outputs this on its P0 port. Both the SRAM data lines and the CPLD are connected to P0. The SRAM uses this connection to send and receive data to and from the microcontroller. The CPLD is programmed to latch the address output on P0 under control of the ALE signal and send the latched address bits to the lower eight address lines of the SRAM.

Meanwhile, the upper eight bits of the address are output on the P2 port of the microcontroller. The 32 KByte SRAM on the XS95 Board uses the lower seven of these address bits while the 128 KByte SRAM on the XS95+ Board gets all eight address bits. The CPLD also receives the upper eight address bits and decodes these along with the PSENB and read/write control line (from pin P3.6 of port P3) from the microcontroller to generate the CEB and OEB signals that enable the SRAM and its output drivers, respectively. Either of the CEB or OEB signals can be pulled high to disable the SRAM and prevent it from having any effect on the rest of the XS95 Board circuitry.

One of the outputs of the CPLD controls the reset line of the microcontroller. The microcontroller can be prevented from having any effect on the rest of the circuitry by forcing the RST pin high through the CPLD. (When RST is active, the microcontroller pins are weakly pulled high.)

Many of the I/O pins of ports P1 and P3 of the microcontroller connect to the CPLD and can be used for general-purpose I/O between the microcontroller and the CPLD. In addition to being general-purpose I/O, the P3 pins also have special functions such as serial transmitters, receivers, interrupt inputs, timer inputs, and external SRAM read/write control signals. If you aren't using a particular special function, then you can use the associated pin for general-purpose I/O between the microcontroller and the CPLD. In many cases, however, you will program the CPLD to make use of the special-purpose microcontroller pins. (For example, the CPLD could generate microcontroller interrupts.) If you want to drive the special-purpose pin from an external circuit, then the CPLD I/O pin connected to it must be tristated.

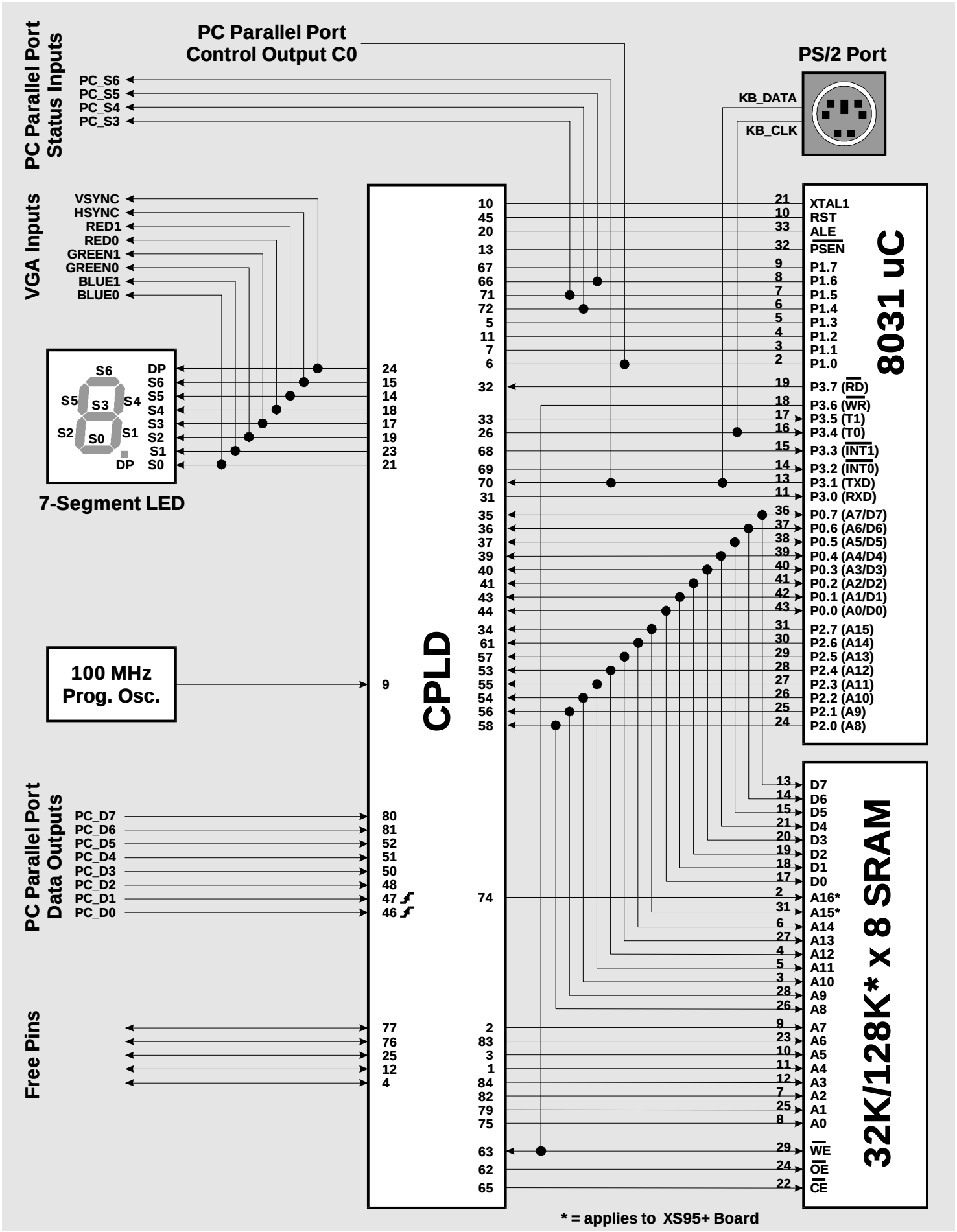
A seven-segment LED digit connects directly to the CPLD. (These same CPLD pins can also drive a VGA monitor.) The CPLD can be programmed so the microcontroller can control the LEDs either through P1 or P3 or by memory-mapping a latch for the LED into the memory space of the microcontroller.

The PC can transmit signals to the XS95 Board through the eight data output bits of the parallel port. The CPLD has direct access to these signals. The microcontroller can also access these signals if you program the CPLD to pass them onto the CPLD I/O pins connected to the microcontroller.

Communication from the XS95 Board back to the PC also occurs through the parallel port. The parallel port status pins are connected to pins of microcontroller ports P1 and P3 . Either the microcontroller or the CPLD can drive the status pins. The PC can read the status pins to fetch data from the XS95 Board.

The CPLD also has access to the clock and data lines of a keyboard or mouse attached to the PS/2 port of the board.

XS95 Pin	Connects to...	Description
21	S0.BLUE0	These pins drive the individual segments of the LED display (S0-S6 and DP). They also drive the color, horizontal, and vertical sync signals for a VGA monitor.
23	S1.BLUE1	
19	S2.GREEN0	
17	S3.GREEN1	
18	S4.RED0	
14	S5.RED1	
15	S6.HSYNCB	
24	DP.VSYNCB	
9	CLK	An input driven by the 100 MHz programmable oscillator.
46	PC_D0	These pins are driven by the data output pins of the PC parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysterisis circuitry.
47	PC_D1	
48	PC_D2	
50	PC_D3	
51	PC_D4	
52	PC_D5	
81	PC_D6	
80	PC_D7	
10	XTAL1	Pin that drives the uC clock input
45	RST	Pin that drives the uC reset input
20	ALEB	Pin that monitors the uC address latch enable
13	PSENB	Pin that monitors the uC program store enable
6	P1.0, PC_C0	These pins connect to the pins of Port 1 of the uC. Some of the pins are also connected to the status input pins of the PC parallel port. The P1.0 port pin of the uC is also connected to the C0 control output from the parallel port.
7	P1.1	
11	P1.2	
5	P1.3	
72	P1.4,PC_S4	
71	P1.5,PC_S3	
66	P1.6,PC_S5	
67	P1.7	
31	P3.0(RXD)	These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated in parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the SRAM. Pins 26 and 70 connect to the clock and data lines of the PS/2 port. Pin 70 connects to a status input pin of the PC parallel port.
70	P3.1(TXD), PC_S6, KB_DATA	
69	P3.2(INTB0)	
68	P3.3(INTB1)	
26	P3.4(T0), KB_CLK	
33	P3.5(T1)	
63	P3.6(WRB), WEB	
32	P3.7(RDB)	
44	P0.0(AD0), D0	These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to the data pins of the SRAM.
43	P0.1(AD1), D1	
41	P0.2(AD2), D2	
40	P0.3(AD3), D3	
39	P0.4(AD4), D4	
37	P0.5(AD5), D5	
36	P0.6(AD6), D6	
35	P0.7(AD7), D7	
58	P2.0(A8), A8	These pins connect to Port 2 of the uC which also outputs the upper address byte. These pins also connect to the upper address bits of the SRAM. Pins 34 and 74 are connected to the 128 KB SRAM address pins only on the XS95+ Board. Pins 34 and 74 do not connect to the 32 KB SRAM on the XS95 Board.
56	P2.0(A9), A9	
54	P2.0(A10), A10	
55	P2.0(A11), A11	
53	P2.0(A12), A12	
57	P2.0(A13), A13	
61	P2.0(A14), A14	
34	P2.0(A15),A15	
74	A16	
75	A0	These pins drive the 8 lower address bits of the SRAM.
79	A1	
82	A2	
84	A3	
1	A4	
3	A5	
83	A6	
2	A7	
62	OEB	Pin that drives the SRAM output enable.
65	CEB	Pin that drives the SRAM chip enable.
4	FREE0	These pins are not connected to other devices and can be used as general purpose I/O.
12	FREE1	
25	FREE2	
76	FREE3	
77	FREE4	


PC Parallel Port
Control Output C0
PS/2 Port
PC Parallel Port
Status Inputs
PC_S6
PC_S5
PC_S4
PC_S3
KB_DATA
KB_CLK
VGA Inputs
VSYNC
HSYNC
RED1
RED0
GREEN1
GREEN0
BLUE1
BLUE0
7-Segment LED
100 MHz
Prog. Osc.
CPLD
8031 uC
XTAL1
RST
ALE
PSEN
P1.7
P1.6
P1.5
P1.4
P1.3
P1.2
P1.1
P1.0
P3.7 (RD)
P3.6 (WR)
P3.5 (T1)
P3.4 (T0)
P3.3 (INT1)
P3.2 (INT0)
P3.1 (TXD)
P3.0 (RXD)
P0.7 (A7/D7)
P0.6 (A6/D6)
P0.5 (A5/D5)
P0.4 (A4/D4)
P0.3 (A3/D3)
P0.2 (A2/D2)
P0.1 (A1/D1)
P0.0 (A0/D0)
P2.7 (A15)
P2.6 (A14)
P2.5 (A13)
P2.4 (A12)
P2.3 (A11)
P2.2 (A10)
P2.1 (A9)
P2.0 (A8)
PC Parallel Port
Data Outputs
PC_D7
PC_D6
PC_D5
PC_D4
PC_D3
PC_D2
PC_D1
PC_D0
32K/128K* x 8 SRAM
D7
D6
D5
D4
D3
D2
D1
D0
A16*
A15*
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
WE
OE
CE
Free Pins
* = applies to XS95+ Board

XS95 and XS95+ Board V1.3 Schematic